

## **REMARKS**

This Amendment is in response to the Office Action dated April 16, 2009 and a telephone interview held on July 21, 2009. Applicant respectfully requests reconsideration and allowance of all pending claims in view of the above-amendments and the following remarks.

### **I. TELEPHONE INTERVIEW**

Applicant's attorney would like to thank the Examiner for the courtesies extended during a telephone interview held on July 21, 2009. During the interview, Applicant's attorney proposed an amendment to independent claim 1, which clarified the steps performed (i.e., "generating" and "transmitting") and clarified that the claimed "dependency pointer" is distinct from any sequence number the stream unit may have. The proposed claim amendment also clarified that the pointer points to at least one other stream unit, as opposed to a sequence number, which is merely a name for the stream unit, itself.

The amendments made above correspond to those discussed in the interview.

#### **A. Support for Proposed Amendments**

Applicant's attorney described the locations in Applicant's original application where support for the amendments can be found.

For example, the dependency pointer is described on page 14, line 15, page 18, lines 14-15, page 20, lines 24-28, page 21, lines 1-4, page 46, lines 15-16, page 48, lines 5-7 and 20-26, and pointers 24-26 shown in FIG. 2.

The sequence number is described on page 30, line 8, page 34, lines 14-15 and page 42, lines 10-29, for example.

As shown and described in the original specification, the dependency pointer and the sequence number are clearly distinct from one another. The specification describes that, if present, the sequence number is continuously increased for each SL packet as a module counter. A discontinuity in the decoder corresponds to one or several missing SL packets. For example, page 14, line 16 and page 46, lines 12-14 explain that the "Marker" value identifies the stream unit itself.

In contrast, as described on page 21, lines 1-4, each stream unit describes in the “dependencyPointer”, the next unit to be received. Page 14, lines 15-16 show that the dependency pointer is distinct from the “Marker”, which identifies the stream unit itself.

According to claim 1, the dependency pointer points to at least one other stream unit of said stream or of another stream that may have been received previously in the terminal, called a required previous unit. Thus, the claimed pointer is distinct from any sequence number that identifies that stream unit, itself, i.e., it points to a different stream unit.

**B. Agreement**

While no specific agreement was reached, the Examiner indicated that the proposed claim amendments would likely overcome the current rejection, but that a further search would need to be performed.

Applicant agreed to file the proposed amendments along with an RCE.

**II. REVIEW OF THE PRESENT APPLICATION**

The present application relates to data transmission in the form of data stream(s). Each data stream is made up of elementary stream units (or packets). An embodiment of the present application optimizes processing of these stream units when they are dependent on preceding stream units in the same stream, or in another stream.

In known techniques, an important difficulty is that of synchronization, when the transmission is made in an asynchronous way. In this case, some stream units emitted after can be received before previously emitted ones. In such a case, one cannot process a received unit of stream, if it is dependent on a preceding unit of stream not yet received.

An embodiment of the present application proposes a new and inventive approach, which includes synchronizing the elementary stream units. The streams themselves might not be synchronized. The synchronization of the streams themselves is realized by the “Object Descriptor” of a stream which is dedicated to the description of the scene.

### III. CLAIM REJECTIONS UNDER 35 U.S.C. §103 BASED ON OKURA AND PUTZOLU

Claims 1-21 are rejected under 35 U.S.C. §103(a) as being allegedly unpatentable over Okura et al., U.S. Publ. No. 2001/0027468, in view of Putzolu et al., U.S. Pat. No. 6,205,140.

Claim 22 is rejected under U.S.C. §103(a) as being unpatentable over Okura et al. in view of Putzolu et al., and further in view of Herrmann et al., U.S. Pat. No. 6,606,329.

#### A. **Okula**

The Office Action acknowledges the primary reference, Okula, does not disclose various elements of claim 1, including a dependency pointer, but relies on a secondary reference, Putzolu et al., for allegedly disclosing these elements.

The Office Action suggests Okura discloses in paragraphs [0125] and [0029] that the SL packet header (stream) has a sequence number and each sequence number has a successive value for a packet and when the scene does not have a successive sequence number, the system determines that there is an abnormality.

Some of the differences between the Applicant's pointers and the sequence numbers of Okura include:

1. The sequence number of Okura does not point to at least one other stream unit of the stream or of another stream.
2. Okura does not disclose a dependency pointer that is distinct from any sequence number the stream unit may have.
3. The sequence number of Okura thus does not correspond to a dependency pointer.
4. The sequence number of Okura is not described in a dependency descriptor.
5. In the present application, the pointer in the current stream unit points to a specific previous required unit. In other words, the pointers are not required to be sequential, as in Okura.
6. With the pointer of claim 1, the procedure allows some stream units to be processed in a different order than transmitted. The pointers allow the current stream unit to be processed if the required previous unit or units (to which the pointer points) have been received, even if other, stream units of the stream that were transmitted between the current stream unit

and the required previous unit have not yet been received. Okura does not disclose or suggest that the sequence numbers of Okura allow a current stream unit to be processed if other stream units of the stream have not yet been received.

7. The pointers point to a specific stream unit or units but does not point to all previous streams units.

8. The sequence number of Okura is the "name" of the stream unit itself, while the pointer of claim 1 is the name of another stream unit of the stream or of another stream, for example.

#### B. Putzolu et al.

Putzolu does not teach or suggest the elements of claim 1 that are not taught or disclosed by Okura.

For example, Putzolu does not teach that "[...] said stream units include at least one pointer that points to at least one other stream unit [...]".

The Examiner considers that the "arrow" of figure 1 and the paragraph (col. 2, lines 63-67) disclose that there is a pointer between stream units.

Applicant disagrees. In column 2, last paragraph, Putzolu discloses "an example of a directed graph illustrating dependencies". Furthermore, it is described that in figure 1, nodes represent a particular media **stream** (col.2, line 58) (audio or video) and not stream units (packets). Moreover, in column 2, lines 63-67, Putzolu discloses that "*an arrow directed from a first node (which is a stream) to a second node (which is also a stream) indicates that the first node (stream) depends directly upon the second node (stream), so that the second node (stream) is required for the first node (stream) to be of use*".

So, in Putzolu, **an arrow is not a pointer**. An arrow just shows a dependency relation between two streams.

Second, Putzolu discloses "stream descriptors" which are enclosed in RAP packets. These RAP packets are occasionally transmitted in a separate control stream, "*for dynamically defining a variable set of media streams belonging to a media presentation, along with their*

*dependencies . . .” (Col. 2, lines 16-19). In column3, lines 33 to 35, Putzolu states, “There is a one-to-one mapping or association between each Stream Descriptor and each media stream comprising the media presentation. Each media stream associated with a Stream Descriptor is implicitly **numbered** according to the numerical order of its associated Stream Descriptor.”*

From col.3, line 62 to col. 4, line 24, Putzolu describes the header 100 of the RAP packet and the stream descriptors 110. More particularly, Putzolu describes that “*Stream Descriptors 110 comprises Dependency Information (d) field **150** . . .*” In col.5, line 28 to 33, it is indicated: “*For a given media stream, **d** field or attribute **150** contains a list of media streams needed for rendering the given media stream. These required media streams may be for technical or social reasons, as discussed earlier. In a preferred embodiment, a media stream that does not depend upon any other media streams will have  $d=0$* ”. No other references can be found in “Putzolu” to describe the form of the list of media stream. More particularly, there is no description of a dependency pointer between stream units in “Putzolu”.

Further, the “Stream Descriptors” of Putzolu, although included in a RAP packet, cannot be considered to disclose a “dependency descriptor” including a “dependency pointer” according to claim 1 since the RAP packet Stream Descriptors do not include a pointer that points to at least one stream unit that is required for processing of that stream unit (i.e., that RAP packet). According to claim 1,

*“at least some of said stream units include at least one pointer that points to at least one stream unit of said stream or of another stream that may have been received previously in the terminal, called a required previous unit, so that the processing of said stream unit is not performed in said terminal if the required previous unit or units have not been received. . . .”*

The RAP packets of Putzolu do not satisfy this feature of claim 1.

### **C. Proposed Combination is Not Obvious**

Even if the one of ordinary skill in the art would have combined Okura and Putzolu, he would not have obtained the limitations of claim 1. Indeed, the disclosure of Putzolu is based on the RTP protocol (*see col.3, line 33*). This protocol is well known to one skilled in the art and is

using sequence numbers simply to indicate that first and second control data packets are consecutive.

To arrive at the invention recited in claim 1, for example, when looking at Okura and Putzolu, it is necessary to:

- Detect the approach of Okura was not adequate;
- Understand that it is desirable to take into account the distinct streams, in order to simplify the processing of decoding;
- Decide to insert pointers in the stream units themselves to allow synchronizing stream units with other stream units of the same stream or another stream previously received. This is not described nor disclosed in Putzolu et al.

Thus the proposed combination is not obvious and, even if made as suggested in the Office Action, fails to teach or suggests the limitations of claim 1.

#### IV. CLAIM REJECTIONS UNDER 35 U.S.C. §103 BASED ON OKURA, PUTZOLU AND HERMANN

Claim 22 is rejected under U.S.C. §103(a) as being unpatentable over Okura et al. in view of Putzolu et al., and further in view of Herrmann et al., U.S. Pat. No. 6,606,329.

Since neither Okura nor Putzolu et al. anticipates the elements of claim 1, and since Hermann also does not disclose the pointers recited in claim 1, the combination of Okura, Putzolu and Hermann also does not teach or suggest the invention recited in claim 1 or the other independent claims.

Hermann is discussed in greater detail in Applicant's prior response, which is incorporated herein.

For the above reasons, Applicant respectfully requests the allowance of all claims and the issuance of a Notice of Allowance.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/  
David D. Brush, Reg. No. 34,557  
900 Second Avenue South, Suite 1400  
Minneapolis, Minnesota 55402-3319  
Phone: (612) 334-3222 Fax: (612) 334-3312

DDB:kmm